# Solving LP Relaxations of Some Hard Problems Is Hard

Daniel Průša and Tomáš Werner

Deptartment of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

Czech Workshop on Applied Mathematics in Engineering (AIME)
February 22, 2018

NP-hard combinatorial optimization problems:

- often expressible as 0-1 LP
- often have natural LP relaxations, which are useful to compute
  - exact solutions for a subclass of the problem
  - bounds for branch-and-cut
  - approximate solutions by rounding schemes
  - parts of optimal solutions (persistency)

Solving LP relaxations:

- in polynomial time (indeed, LP is in P)
- for huge instances hard or impossible, because general LP solvers (simplex, interior point) have
  - high time complexity
  - quadratic space complexity (even for sparse instances)

**Example:** In computer vision (eg, image segmentation), instances of multiway-cut and weighted CSP problems with $\approx 10^7$ variables/constraints.

Find algorithms to solve LP relaxations of particular problems faster and with less memory than general LP solvers!

**Example:** LPs with $\leq 2$ non-zeros in each column of the LP matrix (eg, LP relaxation of weighted CSP with two labels) can be solved by max-flow.

We show that this is often futile because solving LP relaxations of many hard combinatorial problems is not easier than solving the general LP. Precisely:

The general LP problem can be reduced in linear time to the LP relaxations of many combinatorial optimization problems.

I.e., these LP relaxations are LP-complete under linear-time reductions.

**Theorem**

*The general LP problem can be reduced in linear time to the LP relaxation of any of the following problems:*

- *weighted set cover/packing*
- *unweighted set cover/packing*
- *uncapacitated facility location*
- *maximum satisfiability*
- *weighted maximum independent set (clique LP relaxation)*
- *unweighted maximum independent set (clique LP relaxation)*
- *multiway cut*
- *3-D matching*
- *weighted CSP (discrete energy minimization)*

*Moreover, the reductions in red are approximation-preserving.*

[Prusa-Werner PAMI 2015, EMMCVPR 2015, PAMI 2017, SODA 2017]

# Linear Programming

General LP problem:

- $\min\{\, c^T x \mid Ax \geq b,\ x \in \mathbb{R}^n \,\}$ where $a_{ij}, b_i, c_j \in \mathbb{Q}$.
- instance size = number of bits to encode non-zero entries of $A, b, c$

Problem $P$ reduces in linear time to problem $Q$ if: from every instance $p$ of $P$, an instance $q$ of $Q$ can be computed in time $O(\text{size}(p))$ such that

- feasibility of $p$ can be decided from feasibility of $q$ in time $O(\text{size}(p))$,
- from every solution $y$ to $q$, a solution $x$ to $p$ can be computed in time $O(\text{size}(y))$.

# Linear Feasibility Problem in Equality Form (LFE)

Given $A \in \mathbb{Q}^{m\times n}$ and $b \in \mathbb{Q}^m$, solve the system

$$Ax = b$$
$$x \geq 0$$

**Theorem**

*LFE can be reduced in linear time to LFE with binary (0-1) coefficients and at most three variables per equality (LFE-BIN3).*

Proof in three steps:

1. Reduce LFE with rational coefficients to LFE with integer coefficients.
2. Reduce LFE with integer coefficients to LFE with coefficients $\{-1, 0, 1\}$.
3. Reduce LFE with coefficients $\{-1, 0, 1\}$ to LFE-BIN3.

Each term $y = px/q$ (where $p \in \mathbb{Z}$, $q \in \mathbb{N}$) is replaced with $px = qy$.

**Example:** System

$$\begin{aligned} \tfrac{2}{7}x_1 + \tfrac{3}{5}x_2 &= 2, \\ \tfrac{7}{3}x_1 - \tfrac{1}{2}x_2 &= 0 \end{aligned}$$

is replaced with

$$\begin{array}{llll} 2x_1 = 7y_{11}, & 3x_2 = 5y_{12}, & 2 = y_{13}, & y_{11} + y_{12} = y_{13}, \\ 7x_1 = 3y_{21}, & x_2 = 2y_{22}, & & y_{21} - y_{22} = 0. \end{array}$$

## Step 2: From Integer Coefficients to Coefficients $\{-1, 0, 1\}$

Idea: How to construct a term $ax$ for $a \in \mathbb{N}$?

- Construct variables $x_k = 2^k x$ using the system

$$\begin{aligned} x_1 &= x_0 + y_0, & y_0 &= x_0 = x, \\ x_2 &= x_1 + y_1, & y_1 &= x_1, \\ &\vdots & &\vdots \\ x_d &= x_{d-1} + y_{d-1}, & y_{d-1} &= x_{d-1}. \end{aligned}$$

- Then construct $ax$ as the sum of appropriate variables $x_i$.
  **Example:** $11x = x_0 + x_1 + x_3$ because $11 = 2^0 + 2^1 + 2^3$.

System

$$\begin{aligned} 2x_1 + 11x_2 &= 1 \\ 3x_1 - \ \ 6x_2 &= 5 \end{aligned}$$

is transformed to the system

$$\begin{aligned} x_{11} &= x_{10} + y_{10} & y_{10} &= x_{10} = x_1 \\[1em] x_{21} &= x_{20} + y_{20} & y_{20} &= x_{20} = x_2 \\ x_{22} &= x_{21} + y_{21} & y_{21} &= x_{21} \\ x_{23} &= x_{22} + y_{22} & y_{22} &= x_{22} \\[1em] x_{31} &= x_{30} + y_{30} & y_{30} &= x_{30} = 1 \\ x_{32} &= x_{31} + y_{31} & y_{31} &= x_{31} \end{aligned}$$

$$\begin{aligned} x_{11} + (x_{20} + x_{21} + x_{23}) &= x_{30} \\ (x_{10} + x_{11}) - (x_{21} + x_{22}) &= x_{30} + x_{32} \end{aligned}$$

- For $a_{ij}, b_i \in \{0, 1\}$, polyhedron $\{ x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0 \}$ is in $[0, 1]^n$.
- For $a_{ij}, b_i \in \{-1, 0, 1\}$, polyhedron $\{ x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0 \}$ can be unbounded. But:

**Lemma**

*All vertices of polyhedron* $\{ x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0 \}$ *with* $a_{ij}, b_i \in \{-1, 0, 1\}$ *are in the box* $[0, 2^B]^n$ *where*

$$B = \sum_j \left\lceil \log_2 \sum_i |a_{ij}| \right\rceil + \left\lceil \log_2 \sum_i |b_i| \right\rceil.$$

*Moreover, $B$ and the time to compute it are linear in size$(A, b)$.*

1. Scale down input polyhedron $\{ x \in \mathbb{R}^n \mid Ax = b,\ x \geq 0 \}$ such that all its vertices are in the hypercube $[0, \frac{1}{n}]^n$.
   - Replace system $Ax = b$ with $Ax = b\sigma$ where $\sigma = 2^{-\lceil \log_2 n \rceil - B}$.
   - Variable $\sigma$ is constructed similarly as in Step 2.

2. Transform all equations of system $Ax = b$ to the types
   - $x_i = 1$
   - $x_i = x_j$
   - $x_i + x_j = x_k$

   **Example:** $x_1 + x_2 - x_3 + x_4 = 1$ is replaced with $x_1 + x_2 + x_4 = x_3 + 1$, which is replaced with $x_1 + x_2 = x_5$, $x_5 + x_4 = x_6$, $x_3 + x_7 = x_6$, $x_7 = 1$.

3. Transform equations to equations with coefficients $\{0, 1\}$:
   - Equation $x_i = 1$ is already in this form.
   - Equation $x_i = x_j$ is replaced with $x_i + x_k = 1$, $x_j + x_k = 1$.
   - Equation $x_i + x_j = x_k$ is replaced with $x_i + x_j + x_l = 1$, $x_k + x_l = 1$.

# Reducing LFE-BIN3 to LP Relaxation of Set-Cover

Weighted set cover problem:

- Given a collection of $n$ subsets of $m$-element set and a cost $c_j \geq 0$ for each subset.
- Find a sub-collection with minimum cost that covers all $m$ elements.

0-1 LP formulation:

$$\min\{\, c^T x \mid Ax \geq 1,\ x \in \{0,1\}^n \,\}$$

where

$$a_{ij} = \begin{cases} 1 & \text{if } j\text{-th subset contains } i\text{-th element} \\ 0 & \text{otherwise} \end{cases}$$

LP relaxation of set-cover:

$$\min\{\, c^T x \mid Ax \geq 1,\ x \geq 0 \,\}$$

**Theorem**

*Let $c = A^T 1$. Then the optimal value of the LP relaxation is at least $m$, which is attained iff $Ax = 1$.*

**Proof:** $c^T x = 1^T A x \geq 1^T 1 = m$, which is tight iff $Ax = 1$.

Reducing LFE-BIN3 to LP relaxation of set-cover:

- Copy $A$. Set $c = A^T 1$.
- LFE-BIN3 is feasible iff the optimal value of LP relaxation is $m$.
  In that case, every solution to LP relaxation is a solution to LFE-BIN3.